The Pennsylvania State University
The Graduate School

USING ANTS TO FIND COMMUNITIES IN COMPLEX NETWORKS

A Thesis in
Computer Science
by
Mohammad Adi

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2014

The thesis of Mohammad Adi was reviewed and approved* by the following:

Thang N. Bui
Associate Professor of Computer Science
Chair, Mathematics and Computer Science Programs
Thesis Advisor

*Signatures are on file in the Gradute School.

Abstract

Many systems arising in different fields can be described as complex networks, a collection of nodes and edges. An interesting property of these networks is the presence of communities (or clusters), which represents a subset of nodes within the network such that the connections within these nodes are denser than the connections with the rest of the network. In this thesis, we give an ant-based algorithm for finding communities in complex networks. Ants are used to identify edges which are used to assign the nodes into different clusters. Tests on various synthetic and real-world networks show that the algorithm is able to extract the community structure very well and performs well against other algorithms.

Table of Contents

List of Figures

List of Tables

Acknowledgements

¡update later¿

Chapter 1

Introduction

Complex networks are extensively used to model various real-world systems such as social networks, technological (Internet and World Wide Web) networks, biological networks etc. These networks are modeled as graphs where nodes represent the objects in the system and edges represent the relationship among these objects. For example, in a social network, nodes can represent people and two nodes are connected by a link if they are friends with each other.

These networks exhibit distinctive statistical properties. The first property is the "small world effect", which implies that the average distance distance between vertices in a network is short [14]. The second is that the degree distributions follow a power-law [1], and the third one is network transitivity which is the property that two vertices who are both neighbors of the same third vertex, have an increased probabilibty of being neighbors of one another [33].

Another property which appears to be common to such networks is that of community structure (or clustering). While the concept of a community is not strictly defined in the literature as it can be affected by the application domain, one intuitive notion of a community is that it consists of a subset of nodes from the original graph which between them have a higher density of links as compared to their links with the rest of the graph. In this thesis, we describe an ant-based algorithm for detecting communities in graphs.

Over the course of more than a decade, the task of finding communities in networks has received enormous attention from researchers in different fields such as physics,

statistics, computer science etc. As a result, there are currently a vast number of methods which can be used to evaluate the community structure of a network. These methods are described in the next chapter.

Ant algorithms have been previously used to detect communities in graphs [11] [28] [12]. In our approach, we use artificial ants which traverse the graph based solely on local information and deposit pheromone as they travel. This algorithm uses the cumulative pheromone on the edges to build up an initial clustering of the graph. Then a local optimization method is used to reassign the clusters of different nodes based on their degree distribution after which clusters are merged depending on certain rules to obtain the final partitioning of the graph.

The rest of this thesis is organized as follows. Chapter 2 provides more detailed information about the problem statement and covers the previous work done. The ant-based algorithm is described in Chapter 3 and Chapter 4 covers the performance of the algorithm on various synthetic and real-world graphs and compares it to existing algorithms. The conclusion is given in Chapter 5.

Chapter 2

Preliminaries

2.1 Problem Definition

Communities are generally defined to be subsets of vertices which have a high density of links within them. There are various possible definitions of a community and they are divided into mainly three classes: local, global and based on vertex similarity [8] [32]. A more general, quantitative criterion is described in [23] by considering the degree k_i of a node i belonging to a community $S \subset G$, where G is the graph representing the network. The degree of node i can be split as:

$$k_i(S) = k_i^{in}(S) + k_i^{out}(S) \tag{2.1}$$

where $k_i^{in}(S)$ is the number of connections to nodes in its subgraph S and $k_i^{out}(S)$ is the number of connections to nodes outside S. The authors define a community in 2 ways. The subgraph S is a community in the **strong sense** if:

$$k_i^{in}(S) > k_i^{out}(S), \forall i \in S \tag{2.2}$$

The subgraph S is a community in the **weak sense** if:

$$\sum_{i \in S} k_i^{in}(S) > \sum_{i \in S} k_i^{out}(S) \tag{2.3}$$

Even though networks can be directed, undirected, weighted or directed and weighted, we concentrate on undirected and unweighted networks. The problem of community detection can be defined as follows:

Input: An undirected, unweighted graph $G = (V, E)$ where V represents a set of nodes or vertices and E represents a set of edges or links.
Output: A partition $C = \{C_1, \ldots, C_k\}$ of G into k communities where $C_i \cap C_j = \varnothing, i, j = 1, \ldots, k, i \neq j$ and $C_i \subset V, \forall i$.

2.2 Previous Work

The seminal paper by Girvan and Newman [9], resulted in a lot of research into the area of community detection, especially by physicists. As a result, these days there is a wide variety of community detection algorithms from fields like physics, computer science, statistics etc. Covering all of them is beyond the scope of this work, for a more thorough review one can refer the survey by Fortunato [8].

The methods for detecting communities can be broadly classified into hierarchical methods, modularity-based methods and other optimization methods involving statistics or dynamic processes on the graph.

2.2.1 Hierarchical Methods

These type of methods can be further divided into 2 subtypes: divisive hierarchical methods and agglomerative hierarchical methods.

Divisive hierarchical methods start from the complete graph, detect edges that connect different communities based on a certain metric such as edge betweenness [9],

and remove them. Examples of these approaches can be found in [9] [23] [16].

Agglomerative hierachical methods initially consider each node to be in its own community then and merge communities until the whole graph is obtained. Examples can be found in [17] [2] [4].

2.2.2 Modularity-based Methods

Modularity [16] is a metric introduced by Girvan and Newman to evaluate the partitioning of a graph. It is way to quantify the clustering we have obtained in order to determine how good it might be and is a widely adopted quality metric. The idea is that the edge density of the nodes in a cluster should be higher than the expected density of the subgraph whose nodes are connected at random, but with the same degree sequence. This model is called the *null model*.

Using an adjacency matrix representation for the graph, modularity is written as follows:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j) \tag{2.4}$$

where A is the adjacency matrix of the graph G, m is the number of edges in the graph, $\frac{k_i k_j}{2m}$ is the expected number of edges between nodes i and j in the null model and δ is the *Kronecker* functions whose value is 1 if i and j are in the same community and 0 otherwise. Since nodes which do not belong in the same cluster don't contribute towards modularity, it can be rewritten as:

$$Q = \sum_{i=1}^{k} \left(\frac{e_i}{m} - \left(\frac{d_i}{2m} \right)^2 \right) \tag{2.5}$$

where k is the number of communities, e_i is the total number of internal links in cluster i and d_i is the sum of the total degrees of nodes in i. So the first term represents the fraction of the total edges that are in a community and the second term

represents the expected value of the fraction of edges in the null model. Values of Q approaching 1 (which is the maximum), indicate strong community structure [16]. In practice, the value usually ranges from 0.3 - 0.7.

Under the assumption that high values of modularity indicate good partitions, the partition corresponding to its maximum value for a given graph should be the best partition. This is the reasoning employed by modularity-based methods which try to optimize Q to partition a graph. Modularity-based methods are also the most popular methods to be employed for community detection. However, Brandes et. al. showed that maximizing modularity is a NP-hard problem [3], as a result the true maximum of modularity cannot be found in polynomial time unless P = NP. However, there are several algorithms based on different heuristics which approximate the modularity maximimum in a fair amount of time.

The first algorithm to maximize modularity was introduced in [17]. It is an agglomerative clustering approach where vertices are merged based on the maximum increase in modularity. Several other greedy techniques have been developed, some of these can be found in [2] [4] [15] [22] . A simulated annealing approaching to maximizing modularity is described in [10] [13]. Extremal optimization for maximizing modularity was used by Duch and Arenas [7].

Genetic algorithms have also been used for maximizing modularity [30] [19] [18] [20].

2.2.3 Other Methods

Various other techniques for community detection using methods based on statistical mechanics, information theory, random walks etc have been proposed .

Reichardt and Bornholdt [24] proposed a Potts model approach for community detection. Another algorithm based on the Potts model approach is described in [26], this method is fast and it's complexity is a little superlinear in the number of edges in the graph.

Random walks have also been used to detect communities. The motivation behind this is the idea that a random walker will spend a longer amount of time inside a community due to the high density of links within it. These methods are described

in [34] [21] [31].

Information theoretic approaches use the idea of describing a graph by using less information than that encoded in its adjacency matrix. The aim is to compress the the amount of information required to describe the flow of information across the graph. Random walk is used as a proxy for information flow. The minimum description length (MDL) principle [25] can be used as a solution to this problem. The most notable algorithm using this principle, referred to as InfoMap, is described in [27].

2.3 Ant Algorithms

Before describing our algorithm, a review of ant algorithms is given. Ant algorithms are a probabilistic technique for solving computational problems using artificial ants. The ants mimic the behavior of an ant colony in nature for foraging food. As they travel, ants lay down a chemical trail called pheromone, which evaporates over time. The higher the pheromone on a path, the more likely it is to be chosen by the next ant that comes along.

Consider for example a food source and 2 possible paths to reach it, one shorter than the other. Assume two ants set off on both the paths simultaneously. The ant taking the shorter path will return earlier than the other one. Now this ant has covered the trip both ways while the other ant has not yet returned, so the concentration of pheromone on the shorter trail will be more. As a result, the next ant will be more likely to choose the shorter path due to its higher concentration of pheromone. This leads to a further increase of pheromone on that path and eventually all ants will end up taking the shorter path.

Thus, ants can be used for finding good paths within a graph. It is this basic idea that is used in ant algorithms for solving computational problems, but there are different variations. The first such approach, called Ant System (AS), was applied to the Traveling Salesman Problem by Marc Dorigo [6]. In this approach, each ant is used to construct a tour and the pheromone level on all the edges in that tour is

updated based on its length. Each ant picks the next destination based on its distance and the pheromone level on that link. A global update is applied everytime which evaporates the pheromone on all edges so the current best edges would be more like to be chosen by the next ants.

Since in AS each ant updates the pheromone globally, the run time can be quite high. Ant Colony System (ACS) was introduced to address this problem [5]. In ACS, a fixed number of ants are positioned on different cities and each ant constructs a tour, only the iteration best ant, the one with the shortest tour is used to update the pheromone. Ants also employ a local pheromone update in which the pheromone of an edge was reduced as an ant traversed it in order to encourage exploration.

Another variation of AS Max-Min Ant System (MMAS), was introduced by Stutzle and Hoos [29]. The first change in this model is that the pheromone values are limited to the interval $[\tau_{min}, \tau_{max}]$. Secondly, the global update for each iteration is either done by the iteration best ant or the ant which has the best solution from the beginning. This is to used so as to avoid early convergence of the algorithm. Additionally, the pheromone on each edge is initialized to τ_{max} so as to encourage exploration in the beginning of the algorithm. Apart from this, MMAS used the same structure of AS for edge selection and lack of local pheromone update. Both these variations were an improvement over the original AS.

The algorithm proposed here does not fall into the above class of ant algorithms, also called Ant Colony Optimization (ACO) algorithms, instead it falls into the category of Ant-Based Optimization (ABO) methods. While in ACO, ants build complete solutions to the problem, in this approach ants are only used to identify good regions of the search space after which local search or construction methods are used to build the final solution. In ABO, ants only need local information as they traverse the graph. Choosing the next edge involves the pheromone and some heurisitic information based on the rules specified for the ants.

References

[1] Albert-Laszlo Barabasi and Reka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[2] V.D. Blondel, J.L. Guillaume, R. Lambiotte, and E.L.J.S. Mech. Fast unfolding of communities in large networks. *J. Stat. Mech*, page P10008, 2008.

[3] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert G?rke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, 2008.

[4] Aaron Clauset, M. E. J. Newman, , and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, pages 1– 6, 2004.

[5] M. Dorigo and L.M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions on*, 1(1):53–66, 1997.

[6] Marco Dorigo. *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, Italy, 1992.

[7] J. Duch and A. Arenas. Community detection in complex networks using extremal optimization. *Physical Review E*, 72:027104, 2005.

[8] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(35):75 – 174, 2010.

[9] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[10] Roger Guimerà, Marta Sales-Pardo, and Lu´ is A. Nunes Amaral. Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. E*, 70:025101, Aug 2004.

[11] Dongxiao He, Jie Liu, Bo Yang, Yuxiao Huang, Dayou Liu, and Di Jin. An ant-based algorithm with local optimization for community detection in large-scale networks. *CoRR*, abs/1303.4711, 2013.

[12] Di Jin, Dayou Liu, Bo Yang, Carlos Baquero, and Dongxiao He. Ant colony optimization with markov random walk for community detection in graphs. In *Proceedings of the 15th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining - Volume Part II*, PAKDD'11, pages 123–134, Berlin, Heidelberg, 2011. Springer-Verlag.

[13] Claire P. Massen and Jonathan P. K. Doye. Identifying communities within energy landscapes. *Phys. Rev. E*, 71:046101, Apr 2005.

[14] Stanley Milgram. The Small World Problem. *Psychology Today*, 2:60–67, 1967.

[15] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[16] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113, February 2004.

[17] M.E.J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69, September 2003.

[18] C. Pizzuti. A multiobjective genetic algorithm to find communities in complex networks. *Evolutionary Computation, IEEE Transactions on*, 16(3):418–430, 2012.

[19] Clara Pizzuti. Ga-net: A genetic algorithm for community detection in social networks. In Gnter Rudolph, Thomas Jansen, Simon Lucas, Carlo Poloni, and Nicola Beume, editors, *Parallel Problem Solving from Nature PPSN X*, volume 5199 of *Lecture Notes in Computer Science*, pages 1081–1090. Springer Berlin Heidelberg, 2008.

[20] Clara Pizzuti. Boosting the detection of modular community structure with genetic algorithms and local search. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 226–231, New York, NY, USA, 2012. ACM.

[21] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In pInar Yolum, Tunga Gngr, Fikret Grgen, and Can zturan, editors, *Computer and Information Sciences - ISCIS 2005*, volume 3733 of *Lecture Notes in Computer Science*, pages 284–293. Springer Berlin Heidelberg, 2005.

[22] Josep M. Pujol, Javier Béjar, and Jordi Delgado. Clustering algorithm for determining community structure in large networks. *Phys. Rev. E*, 74:016107, Jul 2006.

[23] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2658–2663, 2004.

[24] Jörg Reichardt and Stefan Bornholdt. Detecting fuzzy community structures in complex networks with a potts model. *Phys. Rev. Lett.*, 93:218701, Nov 2004.

[25] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465 – 471, 1978.

[26] Peter Ronhovde and Zohar Nussinov. Local resolution-limit-free potts model for community detection. *Phys. Rev. E*, 81:046114, Apr 2010.

[27] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[28] S. Sadi, S. Oguducu, and A.S. Uyar. An efficient community detection method using parallel clique-finding ants. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–7, 2010.

[29] Thomas Sttzle and Holger H. Hoos. Maxmin ant system. *Future Generation Computer Systems*, 16(8):889 – 914, 2000.

[30] Mursel Tasgin and Haluk Bingol. Community detection in complex networks using genetic algorithm. In *ECCS '06: Proc. of the European Conference on Complex Systems*, April 2006.

[31] Stijn van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000.

[32] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[33] D.J. Watts and S.H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, (393):440–442, 1998.

[34] Haijun Zhou. Network landscape from a brownian particle's perspective. *Phys. Rev. E*, 67:041908, Apr 2003.